Univerzitet u Nišu

Elektronski fakultet

Katedra za računarstvo

Seminarski rad

**Oracle kao distribuirana baza podataka**

****

**Student: Profesor:**

Aleksandar Mladenovic, 883 Doc. Dr Aleksandar Stanimirović

Jun 2021.

Sadržaj

[1. Uvod 3](#_Toc75679767)

[2. Distribuirane baze podataka 4](#_Toc75679768)

[2.1. Prednosti distribuiranih baza podataka 5](#_Toc75679769)

[2.2. Dodatne funkcionalnosti distribuiranih baza podataka 6](#_Toc75679770)

[2.3. Tipovi distribuiranih baza podataka 7](#_Toc75679771)

[2.4. Arhitektura distribuiranih baza podataka 8](#_Toc75679772)

[3. Oracle kao distribuirana baza podataka 8](#_Toc75679773)

[3.1. Arhitektura 10](#_Toc75679774)

[4. Zaključak 13](#_Toc75679775)

[5. Literatura 14](#_Toc75679776)

# Uvod

Tehnologija baza podataka odvela nas je od jednostavne obrade podataka u kojoj svaka aplikacija definiše i održava svoje podatke, do one u kojoj se podaci definišu i njima upravlja centralno. U novije vreme videli smo brzi razvoj mrežnih tehnologija i tehnologija komunikacije podataka, sastavljenih od Interneta, mobilnih i bežičnih uređaja i računarskih mreža. Kombinovanjem ove dve tehnologije, tehnologija distribuirane baze podataka može promeniti način rada iz centralizovanog u decentralizovani. Ova kombinovana tehnologija jedan je od glavnih razvoja na polju baza podataka. Glavna motivacija za razvoj sistema baza podataka je želja da se integrišu operativni podaci organizacije i obezbedi kontrolisani pristup podacima. Razvoj računarskih mreža pomera decentralizovani način rada. Ovaj decentralizovani pristup odražava organizacionu strukturu mnogih kompanija koje se logički distribuiraju gde svaka jedinica odražava svoje operativne podatke. Razmenu podataka i efikasnost pristupa podacima treba poboljšati razvojem distribuiranog sistema baza podataka koji odražava ovu organizacionu strukturu, čini podatke dostupnim u svim jedinicama i čuva podatke u blizini lokacije gde se najčešće koristi. Centralizovanu bazu podataka je lakše održavati, ali ako dođe do zastoja, ceo sistem se sruši i gubimo sve podatke. Distribuirane baze podataka su vrlo korisne u slučajevima kada se dese neke prirodne katastrofe. Zatim, ako imamo centralizovanu bazu podataka, brišu se svi podaci koji se u njoj čuvaju. Danas se to neće dogoditi jer je distribuirani sistem baza podataka zaštićen replikacijom datoteka i deljenjem podataka na više geografski udaljenih lokacija. Korišćenje distribuiranih baza podataka značajno smanjuje troškove primene baze podataka, komunikacije i održavanja. Takve baze podataka imaju zahtevan dizajn i server i teško ih je oporaviti od pada sistema ili velike greške.

# Distribuirane baze podataka

Distribuiranu bazu podataka (DDB) možemo definisati kao kolekciju više logički međusobno povezanih baza podataka distribuiranih preko računarske mreže, a distribuirani sistem za upravljanje bazama podataka (DDBMS) kao softverski sistem koji upravlja distribuiranom bazom podataka, a distribuciju čini transparentnom za korisnika.

Moramo da razlikujemo distribuirane baze podataka od višeprocesorskih sistema koji koriste zajedničko skladište (primarna memorija ili disk). Da bi se baza podataka zvala distribuirana, treba da budu ispunjeni sledeći minimalni uslovi:

* Povezivanje čvorova baze podataka preko računarske mreže - Postoji više računara, koji se nazivaju lokacije ili čvorovi. Ovi čvorovi moraju biti povezane osnovnom komunikacionom mrežom za prenos podataka i naredbi između čvorova
* Logička povezanost baza podataka - Neophodno je da informacije u bazama podataka budu logički povezane
* Odsustvo ograničenja homogenosti među povezanim čvorovima - Nije neophodno da svi čvorovi budu identični u pogledu podataka, hardvera i softvera

Svi čvorovi mogu biti locirani u fizičkoj blizini - recimo, u okviru iste zgrade ili grupe susednih zgrada - i povezane lokalnom mrežom, ili mogu biti geografski raspoređeni na velike udaljenosti i povezani dugim ili širokim područjem mreža. Lokalne mreže obično koriste bežična čvorišta ili kablove, dok mreže na velike daljine koriste telefonske linije ili satelite. Takođe je moguće koristiti kombinaciju mreža.

Mreže mogu imati različite topologije koje definišu direktne puteve komunikacije među čvorovima. Tip i topologija mreže koja se koristi mogu imati značajan uticaj na performanse, a time i na strategije za distribuiranu obradu upita i dizajn distribuirane baze podataka. Za arhitektonska pitanja na visokom nivou, međutim, nije važno koja vrsta mreže se koristi, ono što je važno je da svaki čvor može da komunicira, direktno ili indirektno, sa svakim drugim čvorom. Za efikasan rad sistema distribuiranih baza podataka (DDBS), problemi dizajna mreže i performansi su kritični i sastavni su deo celokupnog rešenja. Detalji osnovne komunikacione mreže su nevidljivi za krajnjeg korisnika.

Koncept transparentnosti proširuje opštu ideju skrivanja detalja implementacije od krajnjih korisnika. Izuzetno transparentan sistem nudi puno fleksibilnosti krajnjem korisniku ili programeru aplikacije, jer zahteva malo ili nimalo svesti o osnovnim detaljima s njihove strane. U slučaju tradicionalne centralizovane baze podataka, transparentnost se jednostavno odnosi na logičku i fizičku nezavisnost podataka za programere aplikacija. Međutim, u DDB scenariju podaci i softver se distribuiraju na više lokacija povezanih računarskom mrežom, pa se uvode dodatne vrste transparentnosti.

Autonomija određuje u kojoj meri pojedinačni čvorovi ili baze podataka u povezanom DDB-u mogu da rade nezavisno. Visok stepen autonomije je poželjan za veću fleksibilnost i prilagođeno održavanje pojedinog čvora. Autonomija se može primeniti na dizajn, komunikaciju i izvršenje. Autonomija dizajna odnosi se na nezavisnost upotrebe modela podataka i tehnika upravljanja transakcijama među čvorovima. Autonomija komunikacije određuje u kojoj meri svaki čvor može da odluči o deljenju informacija sa drugim čvorovima. Autonomija izvršenja odnosi se na nezavisnost korisnika da se ponašaju kako žele.

Pouzdanost i dostupnost su dve najčešće potencijalne prednosti koje se navode za distribuirane baze podataka. Pouzdanost se široko definiše kao verovatnoća da sistem radi u određenoj vremenskoj tački, dok je dostupnost verovatnoća da je sistem neprekidno dostupan tokom vremenskog intervala. Pouzdanost i dostupnost baze podataka možemo direktno povezati sa kvarovima i greškama. Kvar se može opisati kao odstupanje ponašanja sistema od navedenog kako bi se osiguralo pravilno izvršenje operacija. Greške čine onaj podskup stanja sistema koji uzrokuje kvar. Kvar je uzrok greške.

## Prednosti distribuiranih baza podataka

Organizacije pribegavaju upravljanju distribuiranim bazama podataka iz različitih razloga. Neke od važnih prednosti su:

1. ***Poboljšana lakoća i fleksibilnost razvoja aplikacija***. Razvoj i održavanje aplikacija na geografski distribuiranim lokacijama organizacije olakšava se zahvaljujući transparentnosti
2. ***Povećana pouzdanost i dostupnost***. To se postiže izolacijom grešaka na njihovom mestu bez uticaja na ostale baze podataka povezane na mrežu. Kada se podaci i softver DDBMS distribuiraju na nekoliko lokacija, jedna lokacija može propasti dok druge lokacije nastavljaju da rade. Ne mogu se pristupiti samo podacima i softveru koji postoje na neuspeloj veb lokaciji. Ovo poboljšava i pouzdanost i dostupnost. Dalje poboljšanje postiže se razumnim umnožavanjem podataka i softvera na više lokacija. U centralizovanom sistemu, kvar na jednoj lokaciji čini čitav sistem nedostupnim svim korisnicima. U distribuiranoj bazi podataka neki podaci mogu biti nedostupni, ali korisnici će i dalje moći da pristupe drugim delovima baze podataka. Ako su se podaci na neuspešnoj veb lokaciji pre neuspeha replicirali na drugoj, tada to uopšte neće uticati na korisnika.
3. ***Poboljšane performanse***. Distribuirani DBMS fragmentira bazu podataka držeći podatke bliže tamo gde su najpotrebniji. Lokalizacija podataka smanjuje prepirku za CPU i I/O usluge i istovremeno smanjuje kašnjenja u pristupu mrežama širokog područja. Kada se velika baza podataka distribuira na više lokacija, na svakoj lokaciji postoje manje baze podataka. Kao rezultat, lokalni upiti i transakcije pribavljanja podataka na jednoj lokaciji imaju bolje performanse zbog manjih lokalnih baza podataka. Pored toga, svaka lokacija ima manji broj izvršenih transakcija nego ako su sve transakcije predate u jednu centralizovanu bazu podataka. Štaviše, međuupit i unutarupitni paralelizam mogu se postići izvršavanjem višestrukih upita na različitim lokacijama ili razbijanjem upita na određeni broj podupita koji se izvršavaju paralelno. Ovo doprinosi poboljšanju performansi.
4. ***Lakše proširenje***. U distribuiranom okruženju, širenje sistema u smislu dodavanja više podataka, povećanja veličina baze podataka ili dodavanja više procesora je mnogo lakše.

Transparencija dovodi do kompromisa između jednostavnosti upotrebe i opštih troškova obezbeđivanja transparentnosti. Ukupna transparentnost pruža globalnom korisniku pogled na čitav DDBS kao da je jedinstveni centralizovani sistem. Transparentnost se pruža kao dodatak autonomiji, što korisnicima daje strožu kontrolu nad lokalnim bazama podataka. Funkcije transparentnosti mogu se primeniti kao deo korisničkog jezika, koji može prevesti potrebne usluge u odgovarajuće operacije. Pored toga, transparentnost utiče na funkcije koje moraju da pružaju operativni sistem i DBMS.

## Dodatne funkcionalnosti distribuiranih baza podataka

Distribucija baze dovodi do povećane složenosti u dizajnu i implementaciji sistema. Da bi se postigle prethodno navedene potencijalne prednosti, softver DDBMS mora biti u mogućnosti da obezbedi sledeće funkcije pored funkcija centralizovanog DBMS-a:

* ***Praćenje distribucije podataka***. Sposobnost praćenja distribucije podataka, fragmentacije i replikacije proširivanjem DDBMS kataloga.
* ***Distribuirana obrada upita***. Mogućnost pristupa udaljenim lokacijama i prenosa upita i podataka između različitih lokacija putem komunikacione mreže.
* ***Upravljanje distribuiranim transakcijama***. Sposobnost osmišljavanja strategija izvršavanja za upite i transakcije kojima se pristupa podacima sa više lokacija i sinhronizacija pristupa distribuiranim podacima i održavanje integriteta celokupne baze podataka.
* ***Upravljanje repliciranim podacima***. Sposobnost odlučivanja kojoj kopiji replikovane stavke podataka pristupiti i održavanje doslednosti kopija replikovane stavke podataka.
* ***Oporavak distribuirane baze podataka***. Mogućnost oporavka od pojedinačnih padova veb lokacija i od novih vrsta kvarova, poput neuspeha komunikacionih veza.
* ***Bezbednost***. Distribuirane transakcije moraju se izvršiti uz pravilno upravljanje sigurnošću podataka i privilegijama ovlašćenja / pristupa korisnika.
* ***Upravljanje distribuiranim direktorijumom (katalogom)***. Direktorijum sadrži informacije (metapodatke) o podacima u bazi podataka. Direktorijum može biti globalni za čitav DDB ili lokalni za svaku lokaciju. Smještanje i distribucija direktorija su pitanja dizajna i politike.

Te same funkcije povećavaju složenost DDBMS-a u odnosu na centralizovani DBMS. Pre nego što shvatimo sve potencijalne prednosti distribucije, moramo pronaći zadovoljavajuća rešenja za ova pitanja i probleme dizajna. Uključivanje svih ovih dodatnih funkcija teško je ostvariti, a pronalaženje optimalnih rešenja korak je dalje od toga.

## Tipovi distribuiranih baza podataka

Pojam sistem za upravljanje distribuiranim bazama podataka može opisati različite sisteme koji se međusobno razlikuju u mnogim aspektima. Glavna stvar koja je zajednička svim takvim sistemima je činjenica da se podaci i softver distribuiraju na više lokacija povezanih nekim oblikom komunikacione mreže. Možemo razlikovatio brojne tipove DDBMS-a kao i kriterijume i faktore koji čine neke od ovih sistema drugačijim.

Prvi faktor je stepen homogenosti softvera DDBMS. Ako svi serveri (ili pojedinačni lokalni DBMS-ovi) koriste identičan softver, a svi korisnici (klijenti) koriste identičan softver, DDBMS se naziva homogenim; u suprotnom se naziva heterogenim. Drugi faktor povezan sa stepenom homogenosti je stepen lokalne autonomije. Ako ne postoji mogućnost da lokalni čvor funkcioniše kao samostalni DBMS, onda sistem nema lokalnu autonomiju. S druge strane, ako je dozvoljen direktan pristup lokalnih transakcija serveru, sistem ima određeni stepen lokalne autonomije.

U zavisnosti od stepena autonomije susrećemo dva tipa DDBMS-a koji se nazivaju sistem federalnih baza podataka (*federated database system*) i sistem više podataka (*multidatabase system*). U takvim sistemima, svaki server je nezavisni i autonomni centralizovani DBMS koji ima svoje lokalne korisnike, lokalne transakcije i DBA, i stoga ima veoma visok stepen lokalne autonomije. Termin udruženi sistem baza podataka (FDBS) koristi se kada postoji neki globalni prikaz ili šema federacije baza podataka koje aplikacije dele. S druge strane, sistem sa više baza podataka ima punu lokalnu autonomiju u tome što nema globalnu šemu, već interaktivno konstruiše onu po potrebi aplikacije. Oba sistema su hibridi između distribuiranih i centralizovanih sistema.

## Arhitektura distribuiranih baza podataka

Tri alternativna pristupa se koriste za razdvajanje funkcionalnosti kroz različite procese povezane sa DBMS-om. Te alternativne distribuirane arhitekture se nazivaju Client-Server, Collaborating Server i Middleware.

Client-Server sistem ima jedan ili više klijentskih procesa i jedan ili više serverskih procesa, a klijentski proces može poslati upit bilo kome serverskom procesu. Klijenti su odgovorni za probleme korisničkog interfejsa, a serveri upravljaju podacima i izvršavaju transakcije. Dakle, klijentski proces se može pokretati na ličnom računaru i slati upite serveru pokrenutom na glavnom računaru.

Arhitektura klijent-server ne dozvoljava da se jedan upit proteže na više servera, jer bi klijentski proces morao biti sposoban da takav upit razbije u odgovarajuće podupite koji će se izvršiti na različitim lokacijama, a zatim da sastavi odgovore na podupite. Klijentski proces bi zato mogao biti prilično složen i njegove mogućnosti bi se počele preklapati sa serverom, a razlikovanje klijenata od servera postaje teže. Eliminisanje ove razlike vodi nas ka alternativoj arhitekturi, a to je Collaborating Server sistem. Možemo imati kolekciju servera baza podataka, svaki sposoban za pokretanje transakcija protiv lokalnih podataka, koji kooperativno izvršavaju transakcije na više servera.

Arhitektura Middleware je dizajnirana da omogući jednom upitu da obuhvati više servera, bez potrebe da svi serveri baze podataka budu sposobni da upravljaju takvim strategijama izvršavanja na više lokacija. Naročito je atraktivan kada pokušava da integriše nekoliko starih sistema, čije se osnovne mogućnosti ne mogu proširiti

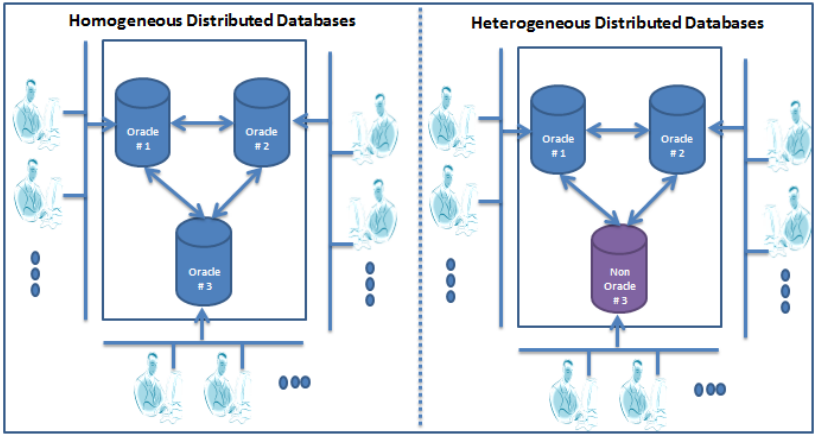
# Oracle kao distribuirana baza podataka

Oracle pruža podršku za homogene, heterogene i klijentsko-serverske arhitekture distribuiranih baza podataka. U homogenoj arhitekturi, najmanje dve Oracle baze podataka nalaze se na najmanje jednoj mašini. Iako su lokacija i platforma baza podataka transparentne za klijentske aplikacije, oni bi morali semantički razlikovati lokalne i udaljene objekte. Koristeći sinonime, ova potreba se može prevazići, tako što korisnici mogu pristupiti udaljenim objektima sa istom sintaksom kao i lokalni objekti. Mogu se koristiti različite verzije DBMS-a, mada mora se napomenuti da Oracle nudi unazad kompatibilnost, ali ne i unapred kompatibilnost između svojih verzija. Na primer, moguće je da neke od SQL ekstenzija koje su ugrađene u Oracle 11i možda neće razumeti Oracle 9.

U heterogenoj arhitekturi, barem jedna od baza podataka u mreži je sistem koji nije Oracle. Lokalna Oracle baza podataka za aplikaciju skriva osnovnu heterogenost i nudi prikaz jedne lokalne, osnovne Oracle baze podataka. Povezivanje se radi korišćenjem protokola kompatibilnog sa ODBC-om ili OLE-DB-om ili komponentama Oracle-ovih heterogenih usluga i Transparentnog mrežnog prolaza. *Oracle Open Gateways* omogućava pristup ne-Oracle bazi podataka sa Oracle servera, koji koristi vezu baze podataka za pristup podacima ili izvršavanje udaljenih procedura u sistemu koji nije Oracle. *Open Gateways* uključuje sledeće:

* ***Distribuirane transakcije***. U okviru dvofaznog mehanizma izvršavanja, transakcije mogu obuhvatati Oracle i ne-Oracle sisteme.
* ***Transparentan pristup SQL-u***. SQL izrazi koje izdaje aplikacija se transparentno transformišu u SQL izraze koje razume sistem koji nije Oracle.
* ***Prolazni SQL i uskladištene procedure***. Aplikacija može direktno da pristupi sistemu koji nije Oracle koristeći verziju SQL-a tog sistema. Pohranjene procedure u sistemu koji nije zasnovan na Oracle SQL-u tretiraju se kao da su PL/SQL udaljene procedure.
* ***Globalna optimizacija upita***. Informacije o kardinalnosti, indeksi i tako dalje u sistemu koji nije Oracle obračunavaju se u Oracle optimizatoru upita servera da bi se izvršila globalna optimizacija upita.
* ***Proceduralni pristup***. Proceduralnim sistemima poput sistema za razmenu poruka ili redova čekanja Oracle server pristupa pomoću poziva PL/SQL udaljenih procedura.

Iz sigurnosne perspektive, Oracle preporučuje da, ako upit potiče sa lokacije A i pristupa lokacijama B, C i D, tada se revizija veza mora vršiti samo u bazi podataka na lokaciji A. To je zato što udaljene baze podataka ne mogu da razlikuju da li uspešni zahtev za povezivanje i sledeći SQL izrazi dolaze sa drugog servera ili lokalno povezanog klijenta.

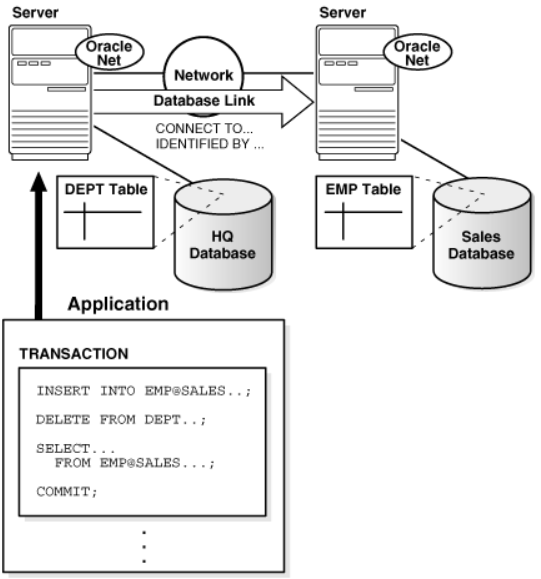


Slika 1. – *Homogena i Heterogena arhitektura*

U arhitekturi klijent-server, sistem baze podataka Oracle podeljen je na dva dela: *front end* kao deo klijenta i *back end* kao deo servera. Klijentski deo je front-end aplikacija baze podataka koja komunicira sa korisnikom. Klijent nema odgovornost za pristup podacima i samo se bavi zahtevanjem, obradom i prezentacijom podataka kojima upravlja server. Deo servera pokreće Oracle i rukuje funkcijama povezanim sa istovremenim deljenim pristupom. Prihvata SQL i PL/SQL izraze koji potiču iz klijentskih aplikacija, obrađuje ih i šalje rezultate nazad klijentu. Oracle klijent-server aplikacije pružaju transparentnost lokacije čineći lokaciju podataka transparentnom za korisnike; Nekoliko karakteristika poput pogleda, sinonima i procedura doprinosi tome. Globalno imenovanje postiže se korišćenjem <TABLE\_NAME @ DATABASE\_NAME> za jedinstveno upućivanje na tabele.

## Arhitektura

Arhitektura Oracle distribuirane baze podataka prikazana je na slici 2. Čvor u sistemu distribuirane baze podataka može da deluje kao klijent, kao server ili oboje, u zavisnosti od situacije. Slika prikazuje dve lokacije na kojima se čuvaju baze podataka pod nazivom HQ (sedište) i Sales (prodaja). Na primeru, aplikaciji koja se pokreće u sedištu, za SQL naredbu nad lokalnim podacima (DELETE FROM DEPT ...), HQ računar deluje kao server, dok za SQL naredbu nad udaljenim podacima (INSERT INTO EMP @ SALES ...), HQ računar deluje kao klijent.



Slika 2. – *Oracle distribuirana baza podataka*

Komunikacija u tako distribuiranom heterogenom okruženju olakšava se putem *Oracle Net Services*, koji podržava standardne mrežne protokole i API-je. U okviru Oracle-ove implementacije distribuiranih baza podataka klijent-server, *Net Services* je odgovoran za uspostavljanje i upravljanje vezama između klijentske aplikacije i servera baze podataka. Prisutan je u svakom čvoru na mreži koji pokreće Oracle klijentsku aplikaciju, server baze podataka ili oba. Pakuje SQL izraze u jedan od mnogih komunikacionih protokola kako bi olakšao komunikaciju od klijenta do servera, a zatim pakuje rezultate natrag klijentu. Podrška koju *Net Services* nude heterogenosti odnosi se samo na specifikacije platforme, a ne na softver baze podataka. Podrška za DBMS, osim Oracle-a, je putem Oracle-ovih heterogenih usluga (*Oracle’s Heterogeneous Services*) i Transparentnog mrežnog prolaza (*Transparent Gateway*). Svaka baza podataka ima jedinstveno globalno ime obezbeđeno hijerarhijskim rasporedom imena mrežnih domena koji se nalazi ispred imena baze podataka da bi ga učinio jedinstvenim.

Oracle podržava veze do baze podataka koje definišu jednosmerni put komunikacije od jedne do druge Oracle baze podataka (*Database Link*). Na primer,komanda *CREATE DATABASE LINK sales.us.americas* uspostavlja vezu sa prodajnom bazom podataka na slici 2. pod mrežnim domenom *us* koji dolazi pod domenom *americas*. Koristeći veze, korisnik može pristupiti udaljenom objektu u drugoj bazi podataka koja je predmet vlasničkih prava, bez potrebe da bude korisnik udaljene baze podataka.

Oracle koristi dvofazni protokol izvršenja da bi se bavio istovremenim distribuiranim transakcijama. Komanda COMMIT pokreće dvofazni mehanizam izvršenja. Pozadinski proces RECO (oporavitelj) automatski rešava ishod onih distribuiranih transakcija u kojima je izvršenje prekinuto. RECO svakog lokalnog Oracle servera automatski izvršava ili vraća sve nesigurne distribuirane transakcije dosledno na svim uključenim čvorovima. Za dugoročne kvarove, Oracle dozvoljava svakom lokalnom DBA da ručno preda ili odbaci bilo kakve sumnjive transakcije i oslobodi resurse. Globalna doslednost može se održati vraćanjem baze podataka na svakoj lokaciji na unapred određenu tačku u prošlosti.

Podaci u Oracle DDBS mogu se kopirati pomoću snimaka ili replikovanih matičnih tabela. Replikacija se pruža na sledećim nivoima:

* ***Osnovna replikacija***. Replikama tabela upravlja se samo za čitanje. Ažuriranjima se mora pristupiti podacima na jednoj primarnoj lokaciji.
* ***Napredna (simetrična) replikacija***. Ovo se širi dalje od osnovne replikacije omogućavajući aplikacijama da ažuriraju replike tabele kroz replicirani DBMS. Podaci se mogu čitati i ažurirati na bilo kojoj veb lokaciji. Ovo zahteva dodatni softver koji se naziva Oracleova napredna opcija replikacije. Snimka stvara kopiju dela tabele pomoću upita koji se naziva upit za definisanje snimka. Jednostavna definicija snimka izgleda ovako: CREATE SNAPSHOT SALES\_ORDERS AS SELECT \* FROM [SALES\_ORDERS@hq.us.americas](mailto:SALES_ORDERS@hq.us.americas);

Oracle grupe prave snimke u osvežavajuće grupe. Određivanjem intervala osvežavanja, snimak se automatski periodično osvežava u tom intervalu za najviše deset procesa osvežavanja snimka (SNP). Ako upit za definisanje snimka sadrži zasebnu ili objedinjenu funkciju, klauzulu GROUP BY ili CONNECT BY ili JOIN ili SET operacije, snimak se naziva složenom snimkom i zahteva dodatnu obradu. Oracle (do verzije 7.3) takođe podržava ROWID snimke koji se zasnivaju na fizičkim identifikatorima redova u glavnoj tabeli.

# Zaključak

Današnji scenariji konkurentnih poslova trebaju da njihova IT preduzeća posluju širom sveta. Ovde baze podataka preduzeća igraju veoma ključnu ulogu, jer su informacije okosnica organizacija. Informacije su raspršene po različitim bazama podataka u obliku transakcionih baza podataka i baza podataka za skladištenje podataka. Generalno je poslovna potreba da imaju jedinstveni konsolidovani prikaz podataka iz baza podataka o transakcijama i skladištenju podataka za proaktivno i brže donošenje odluka na nivou preduzeća i da bi se postigla prednost nad konkurentima. U Oracle okruženjima ove vrste raspršenih baza podataka povezane su putem distribuirane arhitekture baze podataka i veza do baze podataka.

# Literatura

1. Ramez Elmasri and Shamkant Navathe. 2010. Fundamentals of Database Systems (6th. ed.). Addison-Wesley Publishing Company, USA.
2. Raghu Ramakrishnan and Johannes Gehrke. 2002. Database Management Systems (3rd. ed.). McGraw-Hill, Inc., USA.
3. <https://docs.oracle.com/en/database/oracle/oracle-database/index.html>
4. Ghazi Alkhatib and Ronny S. Labban. Transaction Management in Distributed Database Systems: the Case of Oracle’s Two-Phase Commit. Journal of Information Systems Education, Vol. 13(2)
5. <https://en.wikipedia.org/wiki/Distributed_database>